THE EFFECT OF AN INTERDISCIPLINARY SCIENCE COURSE ON STUDENT PERCEPTIONS OF COMPUTER PROGRAMMING

Adam Piggott, Sara Herke, Timothy J. McIntyre, Michael Bulmer

Presenting Author: Sara Herke (s.herke@uq.edu.aul)
School of Mathematics and Physics, The University of Queensland, St Lucia, QLD 4072, Australia

KEYWORDS: interdisciplinary course, attitudes, perceptions, computer programming, science, mathematics

ABSTRACT

Interdisciplinary courses are being offered and recommended by many academic institutions as part of a science degree. In one such first year interdisciplinary science course with a large enrollment we measured attitudes to and perceptions of computer programming at the start and end of semester. For those students with prior computer programming experience, there was a significant positive change in their attitudes to and perceptions of computer programming. The aspects of the course that effected this change were examined. Most students reported that the regular tutorials and the summative assignment, each of which integrated scientific modelling, communication and computer programming, had a positive effect on their attitudes to and perceptions of computer programming. This suggests that an interdisciplinary course can be an effective way to introduce skills such as computer programming.

Proceedings of the Australian Conference on Science and Mathematics Education, The University of Sydney and University of Technology Sydney, 2 - 4 October 2019, pages 162-168, ISBN Number 978-0-9871834-8-4

BACKGROUND

The relationship between students' vocational choices and their attitudes and perceptions of a subject has been the subject of much study (see for example Kennedy, Quinn & Lyons, 2018). These attitudes include students' self-concept of ability, the task value they assign to the subject, and their expectations of success in the subject. A significant body of research also holds that a student's learning can be greatly affected by their attitudes (Else-Quest, Mineo & Higgins, 2013).

The students under consideration in this study are enrolled in SCIE1000 (Theory and Practice in Science) and SCIE1100 (Advanced Theory and Practice in Science). SCIE1000 is an interdisciplinary course which is compulsory for all students majoring in the Bachelor of Science degree at the University of Queensland. SCIE1100 is the companion course to SCIE1000, taken by those students enrolled in the Advanced Science degree; a student taking SCIE1100 has additional assessment related to some additional instruction. We shall hereafter refer to the courses together as SCIE1000/1100. These courses are usually taken in the first semester of the first year of the degree, though SCIE1000 is now offered in semester 2 and summer. SCIE1000 enrolls approximately 1500 students each year and SCIE1100 enrolls approximately 150 students each year.

SCIE1000/1100 is an interdisciplinary course in the sense that it presents "a mixture of science and mathematics although the boundaries of the two disciplines remain visible" (Matthews, Adams & Goos, 2009). The learning objectives for the course relate to the basic creation and use of models in science, fundamental mathematical skills, computer programming, philosophical reasoning, and scientific communication. The interdisciplinary nature of SCIE1000/1100 is also manifest in the diversity of intended majors among the students, and the unusual teaching model employed. The students enrolled in the course come from programs ranging from psychology and biology to physics and mathematics. Most lectures are delivered jointly by two lecturers, a mathematician and a scientist. A four-lecture unit on the philosophy of science is delivered jointly by two philosophers. Students learn basic skills in scientific computing in Python from a computer scientist. Tutorials are staffed by tutors with qualifications from a variety of science, mathematics and computer programming disciplines. The interested reader may consult (Matthews, Adams & Goos, 2009) for a detailed description of the course content and pedagogical approach of SCIE1000/1100.

The first two offerings of SCIE1000, delivered in 2008 and 2009 respectively, were studied in a pair of papers by Matthews, Adams and Goos (2009, 2010). In the first of these papers, the effectiveness of SCIE1000 was studied using a mixed methods approach. The findings of the study included that "students bring strong beliefs about the nature of mathematics and science from secondary school, which can impact significantly on the success of interdisciplinary science–mathematics courses at the tertiary level." We note that SCIE1000 was described as an interdisciplinary science and mathematics course throughout the paper, and the computer programming content was not discussed explicitly even though it can be seen to represent another discipline area. In the second paper, the authors used a quantitative approach to measure the attitudes of biology majors taking the course. In particular, the effect of the course on biology majors' perceptions of computer programming was examined. A key finding was that "SCIE1000 did not contribute positively to gains in appreciation for computing."

The current study is the first output from a larger study with broader aims. The larger study involves investigating attitudes to and perceptions of science, mathematics and computer programming. We evaluate attitudes and perceptions by surveying students using an adaptation of a well-established instrument for measuring attitudinal data, developed by Eccles, Addler and Meece (1984). This instrument has been used in recent years to study attitudes to and perceptions of mathematics and science, see for example Else-Quest, Mineo, and Higgins (2013). We will also be collecting data on how various learning activities affect students' attitudes to and perceptions of science, mathematics and computer programming. Data will be collected both before (in Week 1) and after (in Week 13) the course in each offering of the course, beginning in Semester 1, 2019 and continuing at least until Semester 1, 2021.

This new study was conducted because, while still the same course in aim and spirit, several important details concerning the implementation of SCIE1000/1100 have changed since the work of Matthews et al. (2010). The course became compulsory for students enrolled in a Bachelor of Science degree at the University of Queensland from Semester 1, 2018; new staff have taken over the coordination, lecturing and tutoring of the course; and a new summative programming assignment framed within an interdisciplinary context was introduced in Semester 2, 2018. It is reasonable to ask if the effectiveness of the course remains the same given these changes, and given the inevitable differences between the perceptions and attitudes of those students beginning their tertiary studies in 2008 and those beginning in 2019 and beyond.

In this paper we shall focus our attention on students' attitudes to and perceptions of computer programming using data collected in Semester 1, 2019. This choice is motivated by the contrast between our data and the findings of Matthews et al. (2010) described above.

AIMS

SCIE1000/1100 was designed with the expectation that its interdisciplinary approach would result in improved attitudes to and perceptions of mathematics and computer programming in the sciences. The research questions explored in this paper are natural, given this expectation:

A1. What are students' attitudes to and perceptions of computer programming after taking an interdisciplinary course at the tertiary level?

A2. What role does an interdisciplinary course in science (such as SCIE1000/1100) play in the development of students' attitudes to and perceptions of computer programming?

DESIGN AND METHODS

We use quantitative methods to explore the research questions listed above. In particular, the qualitative research questions are explored through the following quantitative interpretations:

Q1. How do students' attitudes to and perceptions of computing programming, as reported in a quantitative survey instrument, compare to their attitudes to and perceptions of science and mathematics?

Q2. Among those students with prior computer programming experience, is there a measurable change in student attitudes to and perceptions of computer programming before and after SCIE1000/1100?

Q3. Which components of such an interdisciplinary course are the largest contributors to any changing perceptions, as self-reported by the students?

Survey procedures
Students enrolled in SCIE1000/1100 were given an anonymous survey in the first tutorial (Week 1) and last tutorial (Week 13). Almost all of the 1189 students originally enrolled in the course attended their first tutorial, while just over 60% attended their tutorial in Week 13 (students were allowed to miss one tutorial in the semester without penalty). Each student was provided with a participant information sheet and a participant consent form. Survey data were collected using the online survey tool *SurveyMonkey*. The data used in this study are from participants who are 18 years or older and gave informed consent for their data to be used in this study. Human ethics approval was obtained for this project (UQ Human Ethics Approval #2019000164).

The survey included 12 questions on attitudes to and perceptions of computer programming. Each of these questions had options 1-7 for answers, where descriptors were provided for a 1 and a 7, and some questions included a descriptor of a 4 (see Table 1). The questions are based on the instrument developed by Eccles, Addler and Meece (1984). The same 12 questions were also asked for science and for mathematics, where the survey questions in Table 1 had "computer programming" replaced with "science" and "mathematics", respectively.

Table 1: Survey questions on perceptions of computer programming.

Question	**Descriptor of 1**	**Descriptor of 4**	**Descriptor of 7**
How good at **computer programming** are you?	Not good at all	O.K.	Very good
Some students find that they are better at one subject or activity than another. Compared to most of your other activities, how good are you at **computer programming**?	Not as good as other activities	About the same	A lot better than other activities
If you were to list all the students in your class from the worst to the best in **computer programming**, where would you put yourself?	One of the worst	In the middle	The best
How good would you be at learning something new in **computer programming**?	Not good at all	O.K.	Very good
In general, how useful is what you learn in **computer programming**?	Not useful at all		Very useful
Some students find what they learn in one subject or activity more useful than what they learn in another. Compared to most of your other activities, how useful is what you learn in **computer programming**?	Not as useful as what I learn in other activities	About the same	A lot more useful than what I learn in other activities
For me, being good at **computer programming** is	Not important at all		Very important
Some students believe that it is more important to be better at one subject or activity than another. Compared to most of your other activities, how important is it to you to be good at **computer programming**?	Not as important as being good in other activities	About the same	A lot more important than being good in other activities
How much do you like doing **computer programming**?	A little	Some	A lot
Some students find that they like one subject or activity much more than another. Compared to most of your other activities, how much do you like **computer programming**?	Not as much as other activities	About the same	A lot more than other activities
How good do you think you would be in a career requiring **computer programming** skills?	Not good at all	O.K.	Very good
In general, you find working on **computer programming** problems	Very boring	O.K.	Very interesting

The first survey (in Week 1) included the following question before the questions on computer programming: “Have you done any computer programming before?” A student was only presented with the questions on perceptions of computer programming if they answered “Yes” to this question. The final survey (in Week 13) included the questions on perceptions of computer programming for all students taking the survey. The final survey also included questions that asked students to report which components of the course SCIE1000/1100 changed their perceptions of computer programming. They were asked to report positive or negative impact of four key aspects of the course that relate to computer programming: Python contacts, tutorials, the Python and Communication Assignment, and MyPyTutor.

The Python contact is an optional weekly class in which a computer scientist explains the new programming concepts associated with the upcoming tutorial. Less than 30% of students typically attend the Python contacts, but they are recorded using the ECHO360 system and may be viewed at any time by students enrolled in the course. Tutorials focus on further exploration of the mathematical and scientific content and include programming exercises that increase in complexity throughout the duration of the course. Students earn 10% of their final grade by completing pre-work assignments for, and participating fully in, tutorials. The Python and Communication Assignment involves writing an interactive program that simulates echolocation, while explaining the principles of echolocation to users with varying levels of scientific sophistication. The student’s grade on this task determines 15% of their final course grade. MyPyTutor is an online tool with practice programming questions in Python. Use of the tool is optional but highly encouraged. Programming comprehension is assessed on the final exam.

In each survey students were asked non-identifying demographic information (gender, last three digits of phone number, the state in which they completed secondary schooling). This information was used to create an ID code which was then used to link opening and closing survey results. This meant that surveys could be matched even for students who switched tutorials during the semester.

Data analysis

Data were analysed in Excel and R. To measure each student’s attitudes to and perceptions of each discipline — science, mathematics, and computer programming — and their responses (on a scale of 1 to 7) to questions about that discipline were averaged to produce a number, which we call the OAP. The OAP represents the student’s overall attitudes to and perceptions of the discipline. Descriptive statistics, such as the mean and standard deviation of these values across all students, were calculated, and the distributions of these values examined. Descriptive statistics were also considered for various cohorts: those who reported no prior computer programming experience, and those who reported prior computer programming experience.

Descriptive statistics for the Week 1 and Week 13 surveys were compared to determine the change in attitudes to and perceptions of computer programming effected by the course. Given our survey design, this comparison could only be done for the cohort of students who reported prior computer programming experience. Following Matthews et al. (2010), we assess the magnitude of the changes using Cohen’s *d* effect size analysis. In such an analysis, a *d* value of 0.20 is considered indicative of a small effect, 0.50 a medium effect, and 0.80 a large effect (Cohen, 1992).

RESULTS

We identified 634 usable opening surveys from Week 1, from a class with an enrolment of approximately 1200 students. Most of the excluded surveys were from students who were not yet 18 years of age. Of these surveys, 190 (30%) reported having prior computer programming experience and 444 (70%) reported having no prior computer programming experience. The closing survey had 426 usable responses from which there were 136 matching pre-post surveys for students with no prior computer programming experience, and 69 matching pre-post surveys for students with prior experience. Note that in the matching surveys, the proportion of students with prior computer programming experience (34%) is similar to the overall proportion of students with prior experience in the cohort.

To address Q1 we compare the mean OAP for science, mathematics and computer programming for all students. Recall that the majority of students had no prior computer programming experience before this course, so we report the data only for the Week 13 survey. The results are shown in Table 2. We see that students' attitudes to and perceptions of science and mathematics are higher than for computer programming.

Table 3 shows descriptive statistics for overall attitudes to and perceptions of (OAP) computer programming for those students with and without prior computer programming experience in Weeks 1 and 13 (noting that Week 1 data were not obtained for all students).

Table 2: Data on the overall attitudes to and perceptions of (OAP) science and mathematics in Week 13 for all students who completed the closing survey (N=426).

	Mean OAP	Std dev of OAP
Science	5.20	0.91
Mathematics	4.78	1.60
Computer Programming	3.91	1.38

Table 3: Data on the OAP computer programming in Weeks 1 and 13 for those matched students with and without prior computer programming experience.

		Week 1	Week 13
No prior computing (N=136)	Mean OAP	--	3.52
	Std dev of OAP	--	1.24
Prior computing (N = 69)	Mean OAP	4.09	4.82
	Std dev of OAP	1.17	1.35

For those students with prior computing experience, and for whom we can match opening and closing surveys, Table 3 shows a significant change in the mean OAP for computer programming; that is, there was a change from 4.09 to 4.82 in mean OAP. Cohen's d effect size for this change in mean OAP for computer programming is 0.58, which is considered a medium effect. This answers Q2.

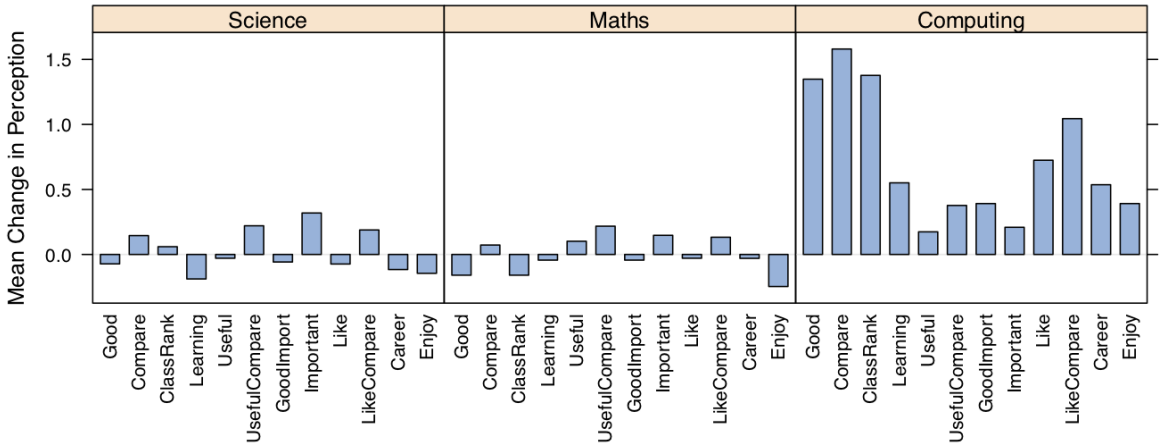

Figure 1: Mean changes for matched students with prior computer programming experience.

Figure 1 shows the mean changes on each of the perception questions (across science, mathematics and computer programming) for the 69 matched students with prior computer programming experience. The biggest effect is in attitudes to and perceptions of computer programming.

Table 4: Data on the effects of course components on perceptions of computer programming.

		Course components			
		Python Contact	Tutorial	P&C Assignment	MyPyTutor
No prior computing	% Positive	33	67	62	33
	% No change	63	23	15	64
	% Negative	4	11	23	3
Prior computing	% Positive	28	70	62	29
	% No change	67	28	28	68
	% Negative	4	3	10	3

To address Q3, we investigated which activities in SCIE1000/1100 effected a change in perceptions of computer programming and the data on these survey questions are given in Table 4. Tutorials had an overwhelmingly positive effect on students' perceptions of computer programming. The Python and Communication Assignment had a more polarizing effect for students with no prior computer programming experience, with 23% of these students reporting a negative impact. This may represent a proportion of students who had not been fully engaged in the activities in tutorials and thus struggled when they were challenged by the nontrivial assessment item. Course tutors reported that some students avoided genuine engagement with programming tasks. It is remarkable that, even among this cohort, 62% of students reported that the assignment promoted a positive change in their perceptions. The majority of students reported that the non-compulsory components of the course (the Python Contact and MyPyTutor) effected no change in their perceptions of computer programming. This result may be due to a lack of engagement with the non-compulsory resources.

CONCLUSIONS

The students' attitudes to and perceptions of computer programming in Week 13 of the course are not as positive as they are for science and mathematics. This is not surprising, given that the majority of the cohort are enrolled in a Bachelor of Science degree and most had no prior computer programming experience. For students who reported prior computer programming experience, there was an increase in their mean overall attitudes to and perceptions of computer programming as measured before and after taking SCIE1000/1100 (Week 1 and Week 13). This is notably different from the findings in Matthews et al. (2010) where the study was limited to biology students. The different findings may be due to different disciplinary interests of the cohorts. The aspects of the course that most improved attitudes towards computer programming, as reported by the students (those with and without prior experience) were the tutorials and the Python and Communication Assignment. Having a significant piece of assessment related to computer programming, but which is framed in the context of an interdisciplinary scientific modelling task, seems to be beneficial.

FUTURE DIRECTIONS

The data in Table 3 show a striking difference between the OAP of computer programming for those students with no prior computer programming experience as compared to those with prior experience. Our study design does not allow us to determine whether or not a disparity between the two groups existed before the course. When conducting this survey in the future, we will ask students with no prior computer programming experience a subset of the computer programming questions in Week 1. When asking students which components of the course changed their perceptions, we will add an option "I did not use that resource" in order to exclude students who did not use a resource.

In the next offering of SCIE1000, some logistical aspects of tutorials have been changed to allow targeted Python interventions for those students with no prior computer programming experience. These interventions include a comprehensive introduction to the optional learning resources, with the ambition to improve engagement.

REFERENCES

Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1), pp.155-159.

Eccles, J., Addler, T., & Meece, J. (1984). Sex Differences in Achievement: A Test of Alternate Theories. *Journal of Personality and Social Psychology,* 46(1): 26-43.

Else-Quest, N. M., Mineo, C. C., & Higgins, A. (2013). Math and science attitudes and achievement at the intersection of gender and ethnicity. *Psychology of Women Quarterly*, 37, 293-309.

Kennedy, J., Quinn, F., & Lyons, T. (2018). The keys to STEM: Australian year 7 students' attitudes and intentions towards science, mathematics and technology courses. *Research in Science Education*, (In Press) https://eprints.qut.edu.au/122151/.

Matthews, K. E., Adams, P., & Goos, M. (2009). Putting it into perspective: mathematics in the undergraduate science curriculum. *International Journal of Mathematical Education in Science and Technology,* 40(7), 891-902.

Matthews, K. E., Adams, P., & Goos, M. (2010). Using the principles of BIO2010 to develop an introductory, interdisciplinary course for biology students. *CBE – Life Sciences Education.* 9, 290-297.

Morphett, A. (2017). Perceptions of mathematics among undergraduate biomedicine students. *Proceedings of the Australian Conference on Science and Mathematics Education, Monash University*, 27-29 September 2017, pages 160-165.